Textbook Treatment of Structured Programming Standards and Guidelines

ABSTRACT: COBOL is still the language of choice in most business computer information systems (BCIS). As a result, COBOL remains the language of choice in structured programming courses required in most undergraduate and graduate computer information system (CIS) and business administration curricula in the United States and Canada. The cost of maintaining COBOL systems, however, can be very expensive for business. A collection of structured programming standards and guidelines (SPSGs) for CIS designed with COBOL has evolved to reduce maintenance costs. This review examines SPSG coverage in textbooks currently used in structured programming courses. These textbooks are evaluated and ranked on the basis of SPSG content. In turn, each SPSG is prioritized according to the degree of inter-textbook coverage. Three textbooks and four SPSGs stand out in the research findings along with suggestions for further empirical research.

Professor Douglas J. Leif
Business Administration Department
Bemidji State University
Bemidji, MN 56601-2699

INTRODUCTION
COBOL In Industry
COBOL is the most widely used business programming language for both current and new business computer information systems (BCIS) (1, 2, 3, 4, 5, 6, 7). Because it is adaptable to client-server computing, personal computer work stations, window environments, computer aided software engineering and re-engineering, graphical user interfaces (GUI), and object oriented programming (OOP), COBOL will likely maintain its popularity (4, 6, 7, 8, 9, 10, 11). The attributes and determinants of COBOL's evolution contribute to it remaining the language of choice in structured programming courses required in most undergraduate and graduate CIS and business administration curricula in the United States and Canada [12, 13, 14, 15, 16].

Structured Programming Standards and Guidelines In Curricula
The approaches to delivering COBOL in curricula however, greatly differ, due in part to the fact that the textbooks most utilized in COBOL courses vary pedagogically. Many textbooks emphasize application development in their COBOL coverage, with individual and unrelated report programs in each chapter. Some textbooks emphasize system development, using related and coordinated data processing programs, not just report programs, throughout many chapters. Several textbooks emphasize program maintainability via structured programming by introducing structured programming techniques in separate textbook units within some chapters. Others emphasize the general design of program logic (flowcharts followed by pseudocode followed by hierarchy charts) prior to coding programs. Even textbooks that have similar emphases in their approach will sequence concepts differently (for example, flowcharting before and after introduction to programming, and control break processing prior to and following sorting).

Meeting Industry Needs of Structured Programming Standards and Guidelines
Research efforts have attempted to assist industry in its constant quest for increasing CIS professional productivity [20]. In a landmark study, structured programming techniques improved maintainability [21]. A collection of structured programming standards and guidelines (SPSGs) have evolved aimed at improving detail design techniques in an effort to reduce BCIS maintenance. SPSGs were intended to generate a more conforming coding style in BCIS, leading to enhanced maintainability and increased programmer productivity. Fifteen SPSGs (Table 1) have been collected, defined, detailed and justified individually elsewhere [22]. The assessment of SPSGs in textbooks must occur to improve instruction of highly structured programming. It should not be ignored because businesses cannot afford to continue to throw 70% to 90% of IS programmer/analyst salary budgets towards the maintenance of legacy BCIS.

This additional investment (without additional expenditures) towards BCIS development would be very important for companies seeking a technological edge over competitors.
SCOPE
Because most CIS professionals learn programming concepts in COBOL courses, the assessment of SPSGs in COBOL textbooks can provide insight into the depth of structured programming as a component of teaching priorities. Thirteen COBOL textbooks (Table 2) were examined for inclusion of each of the 15 SPSGs and ranked accordingly. The SP SGs were then ranked as instructional priorities in the authors' opinions, based on inclusion in the textbooks.

METHODOLOGY
Each textbook was reviewed for any type of coverage of an individual SP SG. The first textbook mention, discussion, example, or illustration of an SP SG was assigned a value of one. The breadth or depth of each SP SG reference, discussion, example, or illustration was not weighted. Therefore, a maximum point total could reach 15 for a textbook that covered every SP SG. A zero value was assigned to each unmentioned SP SG in every textbook. All values were accumulated for each textbook and they were ranked. All values were also totalled for each SP SG and they were ranked accordingly.

RESULTS
Textbook Findings
The Grauer and Villar textbook ranked highest addressing more SP SGs (14) than the other textbooks (Table 3). This textbook delivers more attention to programming structure than the others. Grauer and Villar emphasize the importance of coding accurate applications that are physically attractive, comprehensible, and therefore more easily maintained. The textbook by Horn and Gleason followed by addressing 11 SP SGs. The Spence and Windsor textbook ranked third by addressing 10 SP SGs. Wolff and Feingold covered nine SP SGs, and four textbooks tied for fifth in their treatment of eight SP SGs.

Structured Programming Standards and Guidelines Findings
Four SP SGs receive heavy inter-textbook coverage (Table 3). Indentation, SP SG-4, though not discussed in any narration, was shown in example programs consistently in all 13 texts. SP SG-7, the technique of numbering module prefixes in assigning paragraph names, proved to be the second most popular SP SG incorporated in 12 textbooks, followed by coding one read per input file, SP SG-6, illustrated in 11 books. Strategic spacing, SP SG-3, like indentation, was not addressed in narration but shown in examples in 10 books. SP SG-2 (documentation), SP SG-5 (top-down performance), SP SG-8 (driver module performs), SP SG-10 (cohesive modules), SP SG-11 (loosely coupled modules), and SP SG-12 (avoid commas) each scored seven, eight, or nine. The eight highest ranking textbooks were the only discussants of SP SG-11 (loosely coupled modules).

The quality of textbook SP SG references varied. Both SP SG reference extremes were represented with some textbooks briefly
mentioning an SPSG and others dedicated to an SPG. Also, various SPGMs were exemplified programmatically in textbooks without specific narration. For example, logical spacing (SPG-3) and indentation (SPG-4) were not explained or discussed in the books incorporating them, yet example programs consistently employed the standards. A few books targeted SPGMs in their discussion but illustrated conflicting programming examples. For example, top-down processing (SPG-5) was recommended but example programs illustrated one module performing another module located physically above the calling module [24, 29, 30, 31, 32].

The technique of naming files, their records, and their fields with the same prefix (SPG-1) in a COBOL program provides a great advantage for quickly identifying data definitions, location (source) and purpose. Determining the source and purpose of files, records, and fields may be crucial in improving comprehension (analysis) leading to quicker and more accurate maintenance. Yet employing consistent prefixes (SPG-1) scored only two. Other techniques that received low priority were SPG-9 (avoid literals), SPG-13 (do not use common break processing), and SPG-15 (avoid modules), scoring three each. Only SPG-14 (use cross reference list) scored one, which isn’t surprising because cross reference lists vary due to the fact that they are compiler options dependent upon software vendors.

CONCLUSION

Overall, current textbooks incorporate SPGMs in greatly varying degrees. The textbook by Grauer and Villar was rated the highest by citing 14 of 15 SPGMs. Horn and Gleason was second with 11 SPGMs cited. Spence and Windsor was third with 10 points. Sixty-nine percent of the reviewed textbooks covered at least seven SPGMs. Four SPGMs were thoroughly covered throughout the textbooks: Indentation (SPG-4); number module prefixes (SPG-7); one read per input file (SPG-6); and strategic spacing (SPG-3). Also, authors seem to consistently express or imply their belief that employment of SPGMs positively correlates to reduced maintenance. Because research suggests COBOL’s popularity will continue to spread, and because most CIS professionals receive their initial basic COBOL education in academic programming courses, SPGMs should be incor-
porated at this level. Industry's input on their perception of SPSP priorities would be useful to determine if academia is educating future CIS professionals adequately in SPSP techniques. A marked difference exists in SPSP treatment among textbooks, and presumably CIS application courses. Further research is necessary to align the delivery of SPSPs within COBOL education with industry's use of SPSPs. Also, SPSPs must undergo laboratory research similar to the examination flowcharts have undergone [36]. This will enable us to empirically validate and prioritize individual SPSPs as maintenance improvement tools. Once a refined body of SPSPs is thoroughly tested and accepted, academia can better educate future CIS professionals for industry.

REFERENCES


AUTHOR'S BIOGRAPHY

Douglas Leif is an Assistant Professor of Business Administration at Bemidji State University where he teaches business and computer information system courses. He received his Masters degree specializing in management information systems from Iowa State University. His research interests include detail design in the analysis and design of information systems.

Subscribe!

Subscribe to the Journal of Information Systems Education and you'll get far more than this valuable journal! Because the Journal of Information Systems Education is published by EDSIG, the DPMA's special interest group for education. And as a subscriber, you'll get:

- Four quarterly issues
- Special discounts on EDSIG sponsored activities
- Networking opportunities with others in your profession
- Access to members via Bitnet/Internet
- A forum for discussion of IS teaching and training topics
- The ability to share information on student chapters

Yes! Start my subscription today. I've enclosed a check for $35, payable to the Journal of Information Systems Education, with this completed coupon. (Mail to: John Corrigan, COMPUTERWORLD, 375 Cochituate Road, Framingham, MA 01701.)

Name: ____________________________________________________________
Company or Institution: ____________________________________________
Address: _________________________________________________________
City/State/Zip: _____________________________________________________
Phone: __________________ Fax: __________________ E-mail: ____________

Journal of Information Systems Education
A product from published in 1992
STATEMENT OF PEER REVIEW INTEGRITY

All papers published in the Journal of Information Systems Education have undergone rigorous peer review. This includes an initial editor screening and double-blind refereeing by three or more expert referees.